Final Report

Project Title: Machine Learning to Identify Different Space Weather Conditions – (ID 218)

Name: Rakib Khan

Student number (candidate number): 11227

BEng interim individual project report

Academic year: 3

Supervisor name: Biagio Forte

Word Count: ~6000

Abstract

This report takes a look at the challenge that is applying Machine Learning to the field of Space Weather. It explores the journey of the writer, the different routes and options available, the challenges and problems one may come across when undertaking a project of this nature. While there are many aspects of space weather that can be studied, this report will focus more on the ionosphere, and it’s changing physical conditions and will explore how Machine Learning may be used to better comprehend these conditions and predict future states.

Acknowledgements

I would like to thank my professor and advisor Dr Biagio Forte for his continuous kindness, patience and help during the course of my project. He has been very helpful providing feedback and giving his insights which without this project would not have progressed.

Contents

1 Introduction 6

1.1 Space Weather 6

1.1.1 What is it? 6

1.1.2 As a Natural Hazard 6

1.1.3 Relevancy Today 7

1.1.4 Geomagnetically Induced Currents (GIC) 7

1.1.5 Railway Systems 8

1.1.6 Global Navigation Satellite Systems (GNSS) 8

1.2 Machine Learning 9

1.2.1 Introduction 9

1.2.2 Supervised Learning 10

1.2.3 Unsupervised Learning 11

1.2.4 Reinforcement Learning 11

2 Previous Works 13

2.1 Introduction 13

2.2 Geomagnetic Indices 13

2.3 Kp 13

2.4 Other Focused on Variables 13

2.5 Issues with Past Work 14

3 Approach 15

3.1 Data Acquisition 15

3.1.1 The Need for Data 15

3.1.2 The Search for Data 15

3.1.3 NASA Earth Data 16

3.1.4 UCAR COSMIC Data 17

3.1.5 Total Electron Content File Content and Structure 18

3.2 Selecting the Toolset (Programming Language and Development Environment) 19

3.2.1 Factors to be Considered 19

3.2.2 The Choices 20

3.2.3 Choosing a Development Environment 20

3.2.4 Jupyter Notebook (and it’s issues) 20

3.2.5 The Switch to MATLAB 20

3.3 Data Preparation 21

3.3.1 Preparing the Data 21
3.3.2 ECEF to Geodetic Co-ordinates Example 21
3.3.3 Conversion Example 21
3.3.4 Directory and File Organisation 22
3.4 Data Visualisation 22
3.4.1 Why Visualise? 22
3.4.2 What to Visualise: Total Electron Content (TEC) 22
3.4.3 MATLAB Plots 22
3.4.4 Machine Learning using Regression 24
4 Outcomes / Results 27
5 Milestones 28
6 Uncertainty / Error 29
6.1 Amount of Data used 29
6.2 Uncertainty in the Data 29
6.3 Algorithms and Methods Used 29
7 Further Recommended Work 30
7.1 Repeating with Better Hardware 30
7.2 Using Different Inputs 30
7.3 Using a Different Machine Learning Algorithm 30
7.4 Algorithm Comparison 30
7.5 Forecasting Different Space Weather Variables 30
8 References 31

1 Introduction

In this project, we will take a look at applying machine learning to the field of space weather in order to better understand current conditions and to be able to make predictions and forecasts for future states. Space Weather has been a field which has traditionally been dominated by many classical physics approaches [1], and there is much left to be explored by newer technologies such as machine learning. Therefore, we will dip our feet into this exciting new field and explore it's possibilities. Due to the nature of this project, there are many ways in which one may go about this task. There are many different choices that can be made such as the development environment, programming language, algorithm to be used, type of data, and much more. The time constraints of this project do not allow for the exploration of all these different factors and they may be explored as part of further work. As for now, we will remain flexible in our approach. Firstly, let us clarify some background knowledge.

1.1 Space Weather

1.1.1 What is it?

Many phenomena in space can affect human activities on earth. These phenomena are largely driven by the variability of the sun. It is the study of the variability of the sun on Earth and the various systems surrounding earth, such as the electromagnetic environments on and around earth, that is referred to as Space Weather [2].

1.1.2 As a Natural Hazard

Space weather can be considered as a natural hazard. Activity in the sun can drive changes in the environments on and surrounding earth such as earth's electromagnetic fields, radiation environments in earth's atmosphere, and the upper atmosphere which contains the thermosphere and ionosphere. Changes in these environments can then lead to unwanted disturbances in technologies which are necessary for the functioning of modern day societies. These technologies include significant technologies such as electric power grids and satellite navigation systems [2].

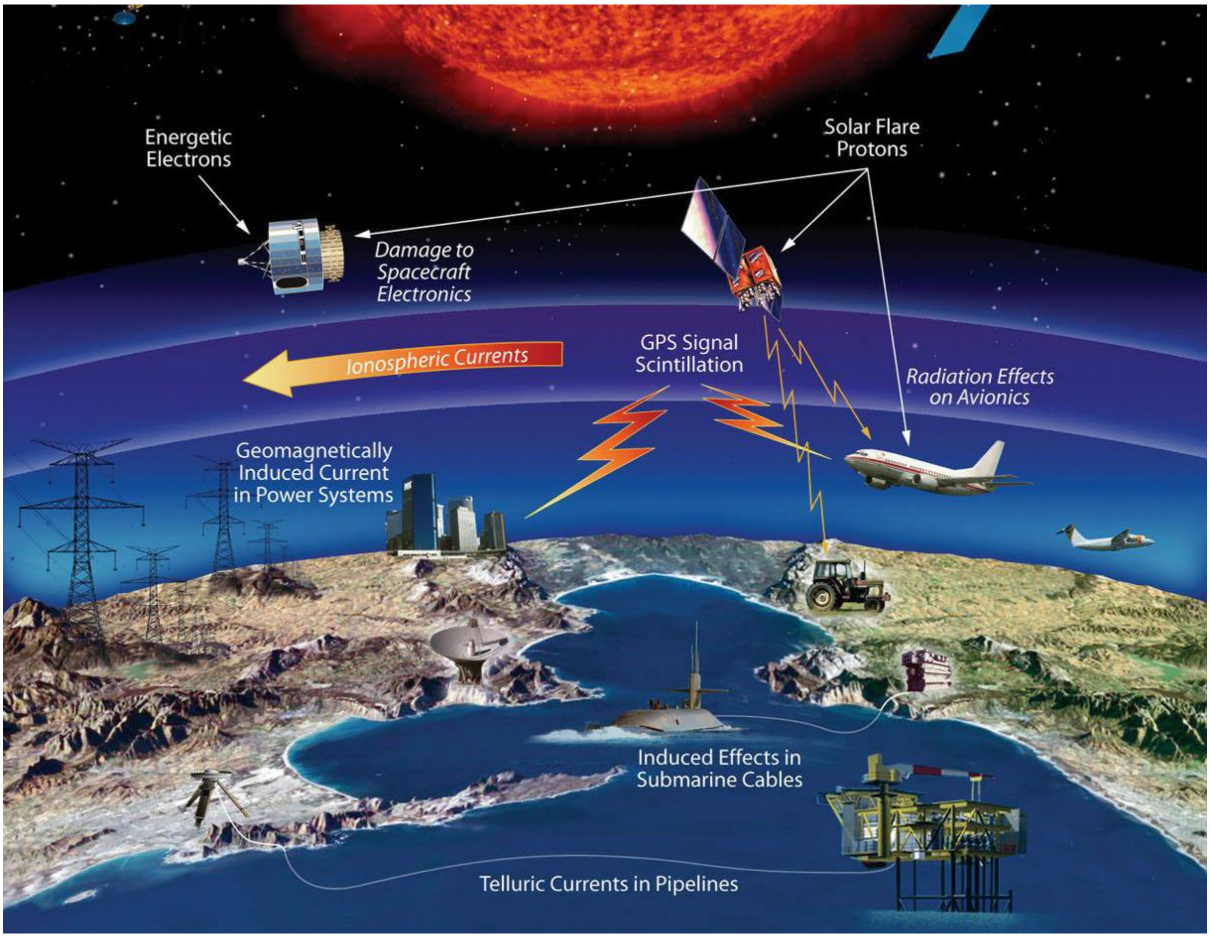

Energetic Electrons
Solar Flare Protons
Damage to Spacecraft Electronics
Ionospheric Currents
GPS Signal Scintillation
Radiation Effects on Avionics
Geomagnetically Induced Current in Power Systems
Induced Effects in Submarine Cables
Telluric Currents in Pipelines

Figure 1: Effects of Impacts of Space Weather

1.1.3 Relevancy Today

These phenomena which collectively are referred to as space weather have been about much longer than we humans have existed on Earth, and we have done nothing about it for so long. So why should we care now? The main reason is because our modern societies have and continue to develop new and advanced technologies which standards of living for countless numbers of people around the globe. However, a good number of these technologies are vulnerable to adverse events brought about by space weather. And as these technologies are disrupted by space weather, so is the lifestyle of many around the globe.

These disturbances have been noted and recorded since as far back as 1847 around the time of the deployment of the first electric telegraph network which was made using metal wires (this is vulnerable to Geomagnetically Induced Currents (GIC)). Even early telephone systems were vulnerable to Geomagnetically Induced Currents and these two technologies were still used until late 20th century.

1.1.4 Geomagnetically Induced Currents (GIC)

Space weather can cause the generation of geoelectric fields within the body of the earth via the process of magnetic induction. These fields can create electric currents which are referred to as geomagnetically induced currents. These currents run through the earth and can also pass through electrically conducting human built structures and cause disturbances. Some significant examples of these structures include electrical power grids, railway circuits, undersea communications cables and much more.

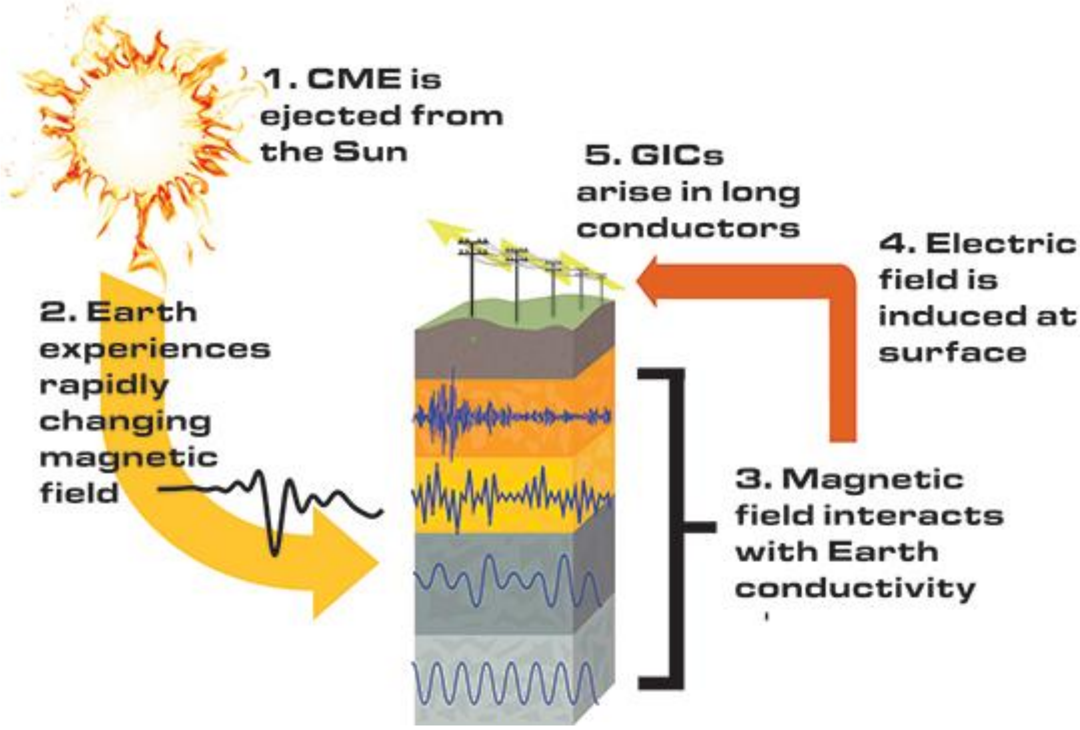

1. CME is ejected from the Sun
5. GICs arise in long conductors
4. Electric field is induced at surface
2. Earth experiences rapidly changing magnetic field
3. Magnetic field interacts with Earth conductivity

Figure 2: Geomagnetically Induced Currents

1.1.5 Railway Systems

As briefly mentioned earlier, space weather can cause the creation of geomagnetically induced currents which can then flow into railway systems. Track circuits within these systems have important jobs including the location of trains and the controlling of the switching the colours of the track signal lights. There have been recorded cases of these tasks being failed to be performed due to space weather. Signal lights have been observed to switch incorrectly and the malfunctioning is thought to have occurred due to these geomagnetically induced currents.

This can have extremely serious consequences as human lives are at risk.

1.1.6 Global Navigation Satellite Systems (GNSS)

GNSS is an example of a major technology that is highly vulnerable to the impacts of space weather. Satellite navigations provides features such as providing precise location and accurate timing, an invention that has become deeply embedded in modern society and something that we heavily rely on for many important services. Activities such as trading which is of the utmost economic importance use GNSS for highly accurate timing to the millisecond in order to time stamp transactions. And of course, GNSS is used for many navigational purposes such as in aircraft, shipping, road transport and much more.

This works by satellite receivers detecting signals from a number of satellites and then analysing those signals and using the times it took for them to travel to work out their locations. One assumption made in this process is that the signals travel at the speed of light.

This is where space weather events can start to create problems. These signals pass through the ionosphere, and space weather events can change the state of the ionosphere. Furthermore, the speed of the signals is affected by the current state of the ionosphere. Even a time delay caused by this on the small scale of tens of nanoseconds can lead to errors on the scale of several meters. This is actually a rather large error which governments around the world have invested billions of dollars to correct. Examples of error correction systems include the WAAS from the USA, EGNOS from Europe, MTSTAT from Japan. In extreme cases of space weather, these GNSS services can go down for multiple days further highlighting the seriousness of the impacts of space weather.

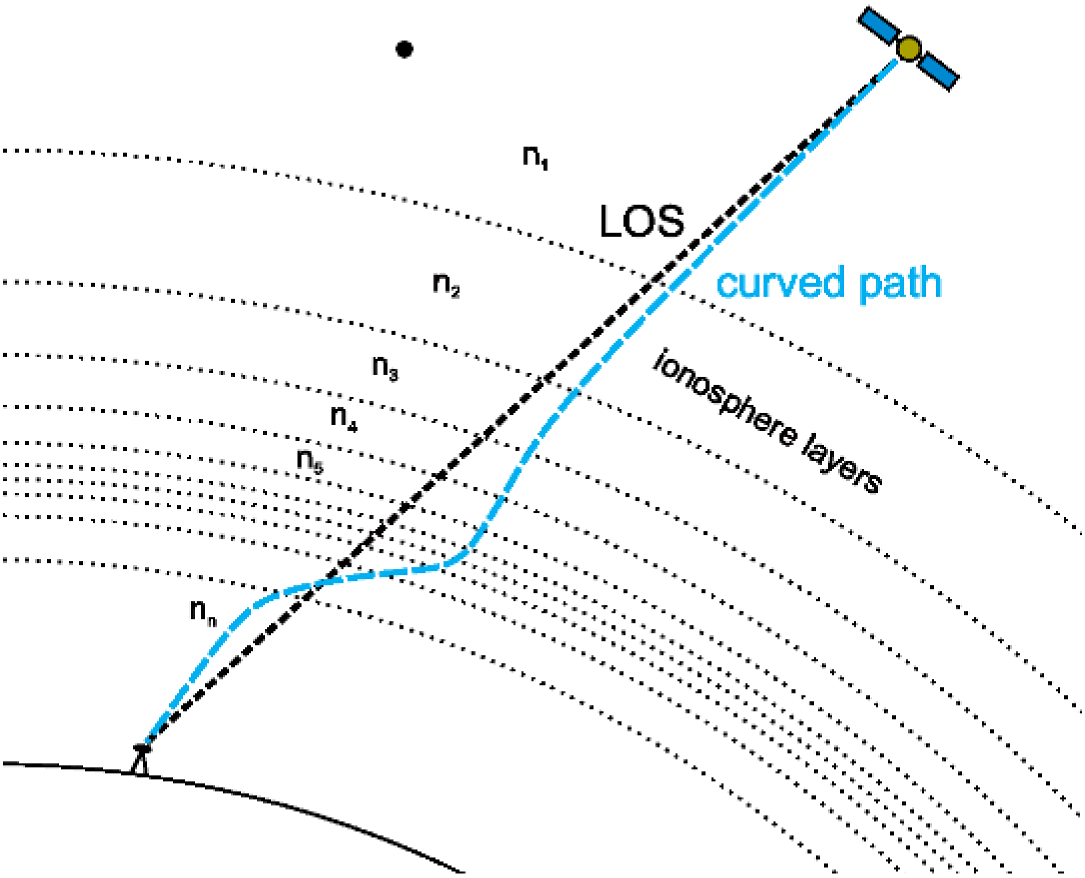


Figure 3: Signals travelling through Ionosphere

1.2 Machine Learning

1.2.1 Introduction

Machine Learning is a new technology which has been growing faster and faster in recent years being used in almost all fields of science and engineering [3]. Machine learning can be described as the set of tools that allow us to teach computers to perform certain tasks without explicitly programming them on how to complete those tasks. In the context of this project, it is a set of methods and algorithms that can be used for many problems such as forecasting (making predictions for continuous data), classification of data, making sense of data that is not well understood, exposing undiscovered relationships between several variables, and much more.

There are a few different types of machine learning. We will take a brief look at three different popular types which include supervised learning, unsupervised learning, and reinforcement learning.

1.2.2 Supervised Learning

Supervised learning is where the programs learns by using a lot of example data also known as training data. The training data consists of inputs which map to certain outputs. The program will try to learn patterns between the inputs and outputs. The goal is for the program to produce the correct output when given a new input. This output can be a value (regression) or a class (classification).

During training, the program is being 'supervised', it is being told what the answer should be given a certain input. After the program has been trained, it can be tested with a new set of data which is also known as test data. This makes this kind of machine learning useful for tasks such as forecasting (as we are doing so in this project). The program is given a new set of inputs which are unlabelled, and the program will spit out outputs which it believes the inputs map to.

This process of training and testing can be repeated over and over to improve the programs accuracy. In the case of space weather forecasting, we would use past available data which would serve as the training data. Then we would feed in future data values and hope that the program would be able to accurately predict them.

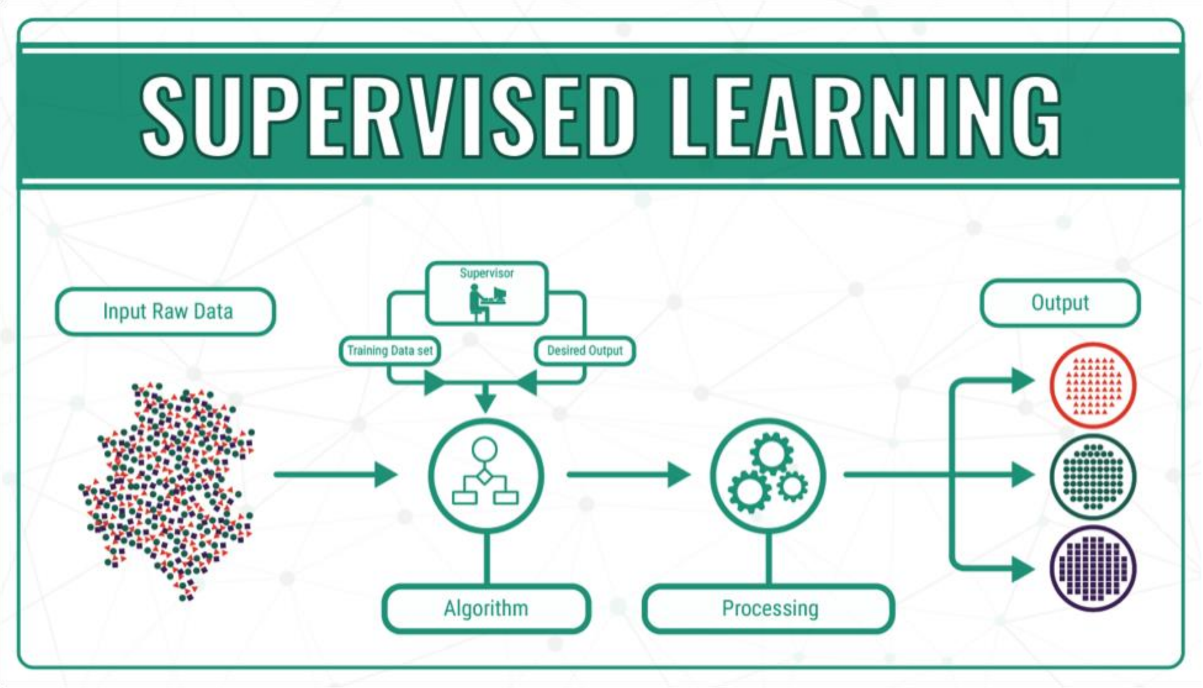


Figure 4: Supervised Learning

1.2.3 Unsupervised Learning

Unsupervised learning is the opposite to supervised learning. The program is not given labelled data, there are no ‘correct’ answers. Therefore, the program is learning unsupervised. In this kind of learning, the program receives data and attempts to find interesting patterns and trends. A lot of new discoveries and concepts can be made this way.

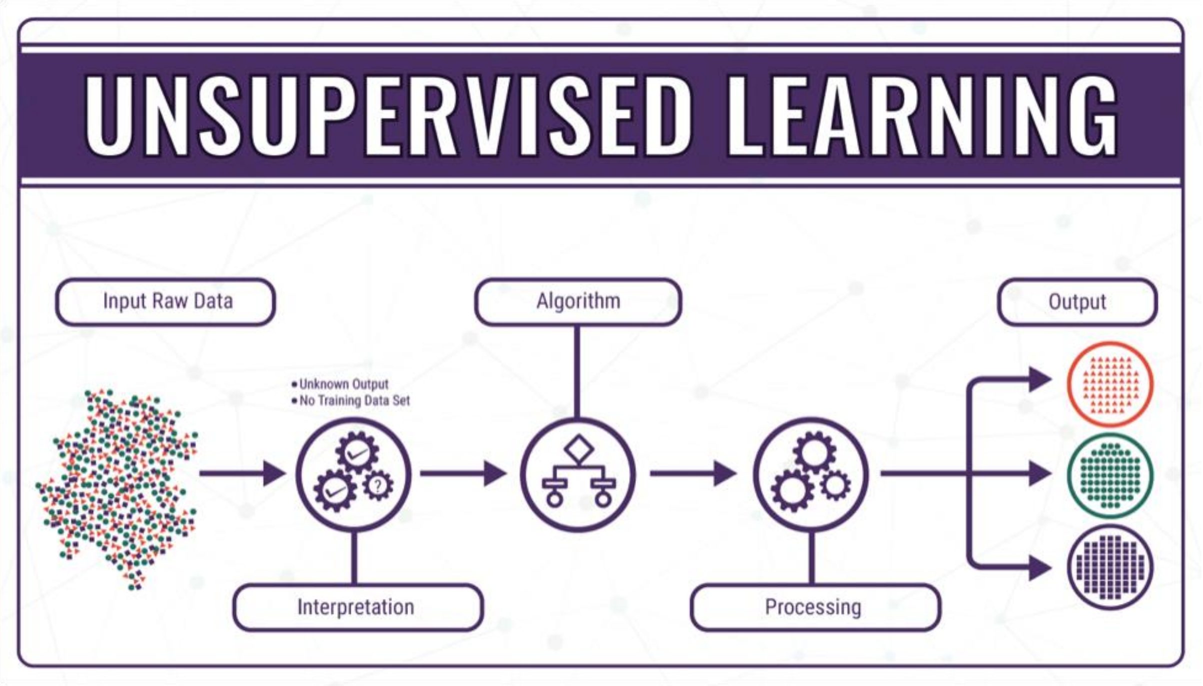


Figure 5: Unsupervised Learning

1.2.4 Reinforcement Learning

In order for us to create supervised learning programs, we be able to clearly define correct answers for the program to learn from. However there are times when we simply do not know the correct answer for all situations. For example, if our goal was to train a robot to walk, we wouldn’t know whether or not each step the robot took was ‘correct’. This is where reinforcement comes in handy.

In reinforcement learning, there is an ‘agent’ which learns and there is also a reward function. This helps the agent know when it is performing well and when it is failing to learn effectively.

This kind of machine learning finds it’s use cases in many different kinds of fields such as in the creation of autonomous vehicles, factory control, and even things such as robots which have been shown to beat world champions in games such as backgammon [4].

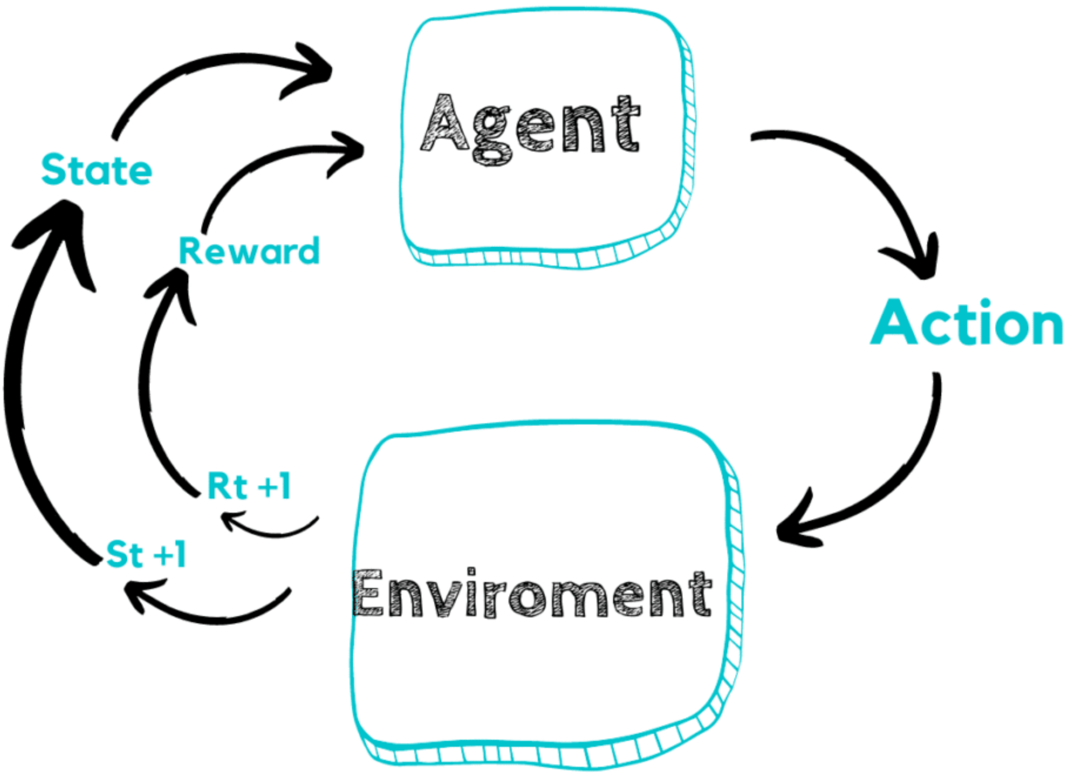


Figure 6: Reinforcement Learning

2 Previous Works

2.1 Introduction

It is not the first time that someone will have made an attempt to integrate machine learning with the field of space weather. There have been many previous works which take a look at machine learning (neural networks in particular) to make predictions of quantities such as geomagnetic indices since the 1990s [1]. We will take a look at three areas which have commonly been the focus of many machine learning and space weather projects in the past. These are the geomagnetic index prediction, relativistic electron prediction, and solar eruption prediction.

2.2 Geomagnetic Indices

Geomagnetic index prediction has been one of the most popular looked at tasks in using machine learning with space weather. Geomagnetic indices are measures of geomagnetic activity. They capture information about the state of the magnetosphere condense it into a single number. Many of these indices exist, each describing a different aspect of geomagnetic activity. Examples include …. With the most popular ones used for prediction tasks being Kp, Dst

2.3 Kp

The Kp index is an indicator of disturbances that are present in the earth's magnetic field. It is used as a tool by the Space Weather Prediction Center when deciding whether to issue geomagnetic warnings and alerts.

There have been many suggestions in the past to use neural networks in order to predict Kp values in the future (one or more hours ahead of time) [5]. And many real forecasts have been created based of these models and are currently running at:

- RWC, Sweden - http://www.lund.irf.se/forecast/kp/
- Rice Space Institute, USA - http://mms.rice.edu/mms/forecast.php
- INPE, Brazil - http://www2.inpe.br/climaespacial/portal/swd-forecast/
- Space Environment Prediction Center, China - http://www2.inpe.br/climaespacial/portal/swd-forecast/

US Space …. Provided a model from …. To …. Based on the wing model….this model compared to others….. produced better results due to larger training set and due to using the current Kp value as an input val …

2.4 Other Focused on Variables

The Dst index is used to measure the deviance of the Earth's magnetic field's horizontal component from it's long term average. It is an hourly based index which is measured in nano tesla. This has also been the focus of many past works.

Solar flares and relativistic electrons have also received their fair amount of attention in past works, with attempts of using neural networks to create forecasts for them

2.5 Issues with Past Work

There is an issue that it is hard to know whether or not research has truly been progressing in this field. This can be attributed to the fact that there is a huge amount of freedom in these projects and lots of paramerers that can be played around with. Hence, projects have not necessarily built off of each other and have gone on their own directions making them hard to compare and assess.

3 Approach

3.1 Data Acquisition

3.1.1 The Need for Data

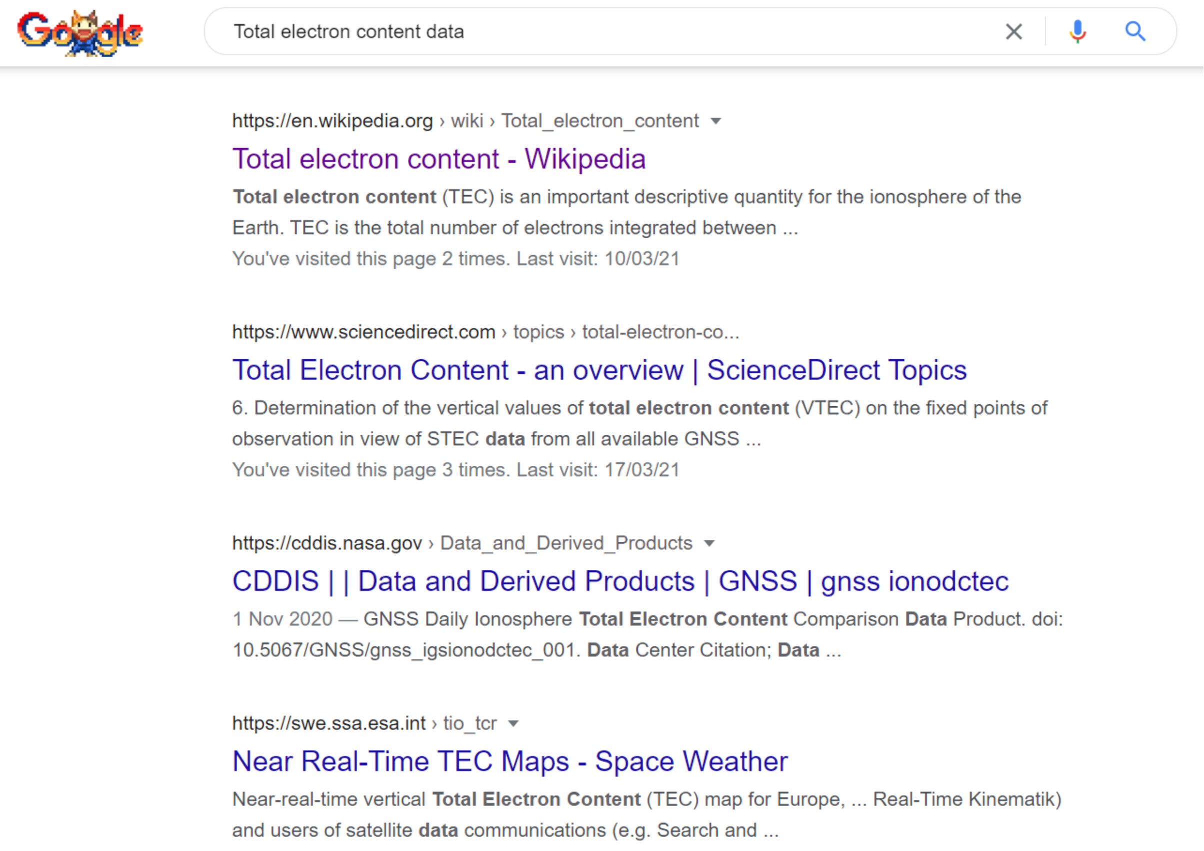
At the very core of Machine Learning lies data. Without it, the program can not learn and therefore can not make the forecasts that we seek. It is necessary to have good quality data in order for the program to run as well as possible.

However, an issue arises. It can be difficult to locate data when the data is not popularly available. We are looking for data concerning Total Electron Content in the ionosphere. This is not a particularly popular area of research especially in comparison to other fields of research. Therefore, it does not have the same quantity and ease of access of data.

Another issue to consider when acquiring data is the format of the data. Not all formats of data (the file type etc) are fit for use in machine projects like these. It is necessary to take into consideration whether the type of data and the file type will be usable in whatever development environment we choose to set up.

3.1.2 The Search for Data

Searching for data as predicted proved to be a rather cumbersome task. Googling does not bring back a wide variety of results, and even sites like GitHub do not have a lot of results to choose from.



Total electron content data

https://en.wikipedia.org › wiki › Total_electron_content

Total electron content - Wikipedia

Total electron content (TEC) is an important descriptive quantity for the ionosphere of the Earth. TEC is the total number of electrons integrated between ...
You've visited this page 2 times. Last visit: 10/03/21

https://www.sciencedirect.com › topics › total-electron-co...

Total Electron Content - an overview | ScienceDirect Topics

6. Determination of the vertical values of **total electron content** (VTEC) on the fixed points of observation in view of STEC **data** from all available GNSS ...
You've visited this page 3 times. Last visit: 17/03/21

https://cddis.nasa.gov › Data_and_Derived_Products

CDDIS | | Data and Derived Products | GNSS | gnss ionodctec

1 Nov 2020 — GNSS Daily Ionosphere **Total Electron Content** Comparison **Data** Product. doi: 10.5067/GNSS/gnss_igsionodctec_001. **Data** Center Citation; **Data** ...

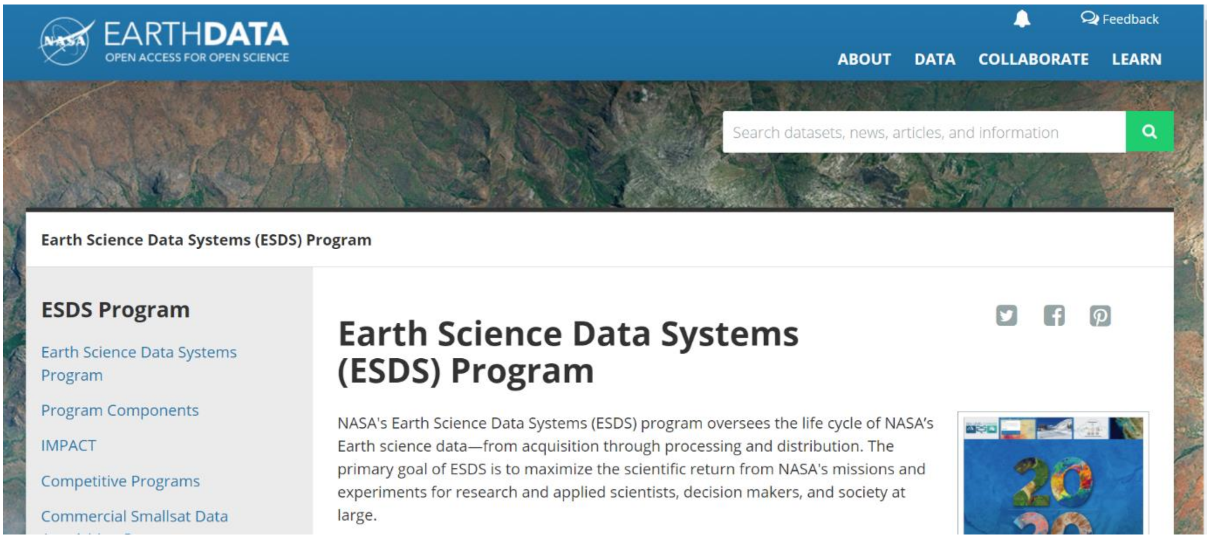
https://swe.ssa.esa.int › tio_tcr

Near Real-Time TEC Maps - Space Weather

Near-real-time vertical **Total Electron Content** (TEC) map for Europe, ... Real-Time Kinematik) and users of satellite **data** communications (e.g. Search and ...

Figure 7: Google Search Results for TEC

3.1.3 NASA Earth Data


EARTHDATA
OPEN ACCESS FOR OPEN SCIENCE
Feedback
ABOUT
DATA
COLLABORATE
LEARN
Search datasets, news, articles, and information

Earth Science Data Systems (ESDS) Program

ESDS Program

Earth Science Data Systems Program

Program Components

IMPACT

Competitive Programs

Commercial Smallsat Data

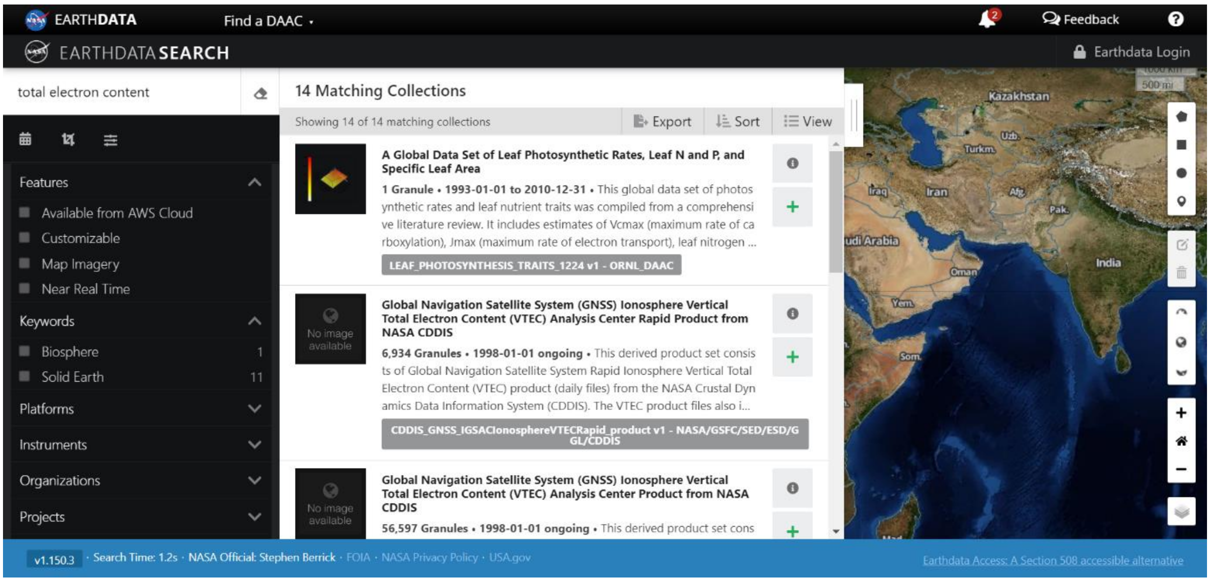
Earth Science Data Systems (ESDS) Program

NASA's Earth Science Data Systems (ESDS) program oversees the life cycle of NASA's Earth science data—from acquisition through processing and distribution. The primary goal of ESDS is to maximize the scientific return from NASA's missions and experiments for research and applied scientists, decision makers, and society at large.

Figure 8: Earth Data Home Page

One promising site that comes up upon search however is NASA Earth Data [6]. NASA Earth Data is a site owned by NASA which provides scientific data collected by NASA for free to be used by the general public. As NASA puts it, the goal of NASA Earth Data is to “maximize the scientific return from NASA's missions and experiments for research and applied scientists, decision makers, and society at large”.

However, the interface can be unfriendly to beginners and the desired data can be difficult to locate.



total electron content

Features

- Available from AWS Cloud
- Customizable
- Map Imagery
- Near Real Time

Keywords

- Biosphere 1
- Solid Earth 11

Platforms

Instruments

Organizations

Projects

14 Matching Collections

Showing 14 of 14 matching collections

A Global Data Set of Leaf Photosynthetic Rates, Leaf N and P, and Specific Leaf Area

1 Granule • 1993-01-01 to 2010-12-31 • This global data set of photosynthetic rates and leaf nutrient traits was compiled from a comprehensive literature review. It includes estimates of Vcmax (maximum rate of carboxylation), Jmax (maximum rate of electron transport), leaf nitrogen ...

LEAF_PHOTOSYNTHESIS_TRAITS_1224 v1 - ORNL_DAAC

No image available

Global Navigation Satellite System (GNSS) Ionosphere Vertical Total Electron Content (VTEC) Analysis Center Rapid Product from NASA CDDIS

6,934 Granules • 1998-01-01 ongoing • This derived product set consists of Global Navigation Satellite System Rapid Ionosphere Vertical Total Electron Content (VTEC) product (daily files) from the NASA Crustal Dynamics Data Information System (CDDIS). The VTEC product files also i...

CDDIS_GNSS_IGSAClonosphereVTECRapid_product v1 - NASA/GSFC/SED/ESD/GGL/CDDIS

No image available

Global Navigation Satellite System (GNSS) Ionosphere Vertical Total Electron Content (VTEC) Analysis Center Product from NASA CDDIS

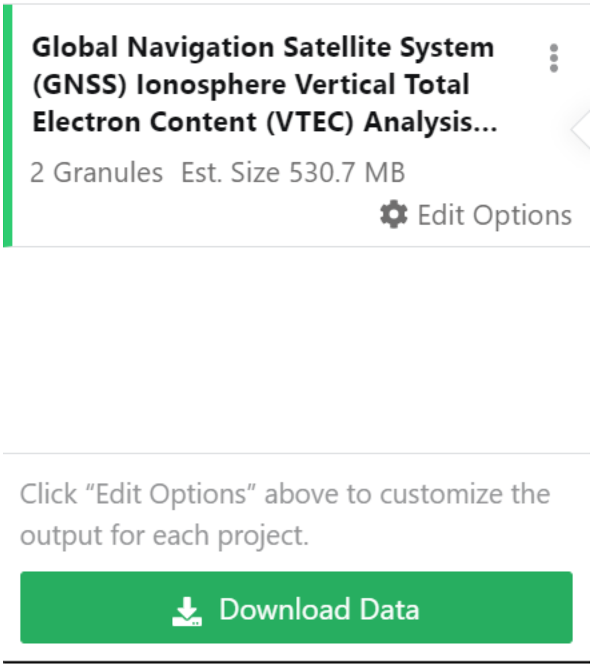
56,597 Granules • 1998-01-01 ongoing • This derived product set cons

v1.150.3 · Search Time: 1.2s · NASA Official: Stephen Berrick · FOIA · NASA Privacy Policy · USA.gov

Earthdata Access: A Section 508 accessible alternative

Figure 9: Earth Data Search Results

Data sizes are also very large. A mere two granules takes up more than half a gigabyte. This is not practical for workstations with limited IT capabilities such as the one being used in this project.



Global Navigation Satellite System (GNSS) Ionosphere Vertical Total Electron Content (VTEC) Analysis...

2 Granules Est. Size 530.7 MB

Click “Edit Options” above to customize the output for each project.

Figure 10: Granules of Data

3.1.4 UCAR COSMIC Data

COSMIC

CONSTELLATION OBSERVING SYSTEM FOR METEOROLOGY, IONOSPHERE, AND CLIMATE

DEVELOPING INNOVATIVE GLOBAL NAVIGATION SATELLITE SYSTEMS OBSERVATIONAL TECHNIQUES FOR THE EARTH SCIENCE COMMUNITY

Figure 11: UCAR COSMIC Home Page

This is another site that I came across in the search for data and is the site that became the primary source of data for this project.

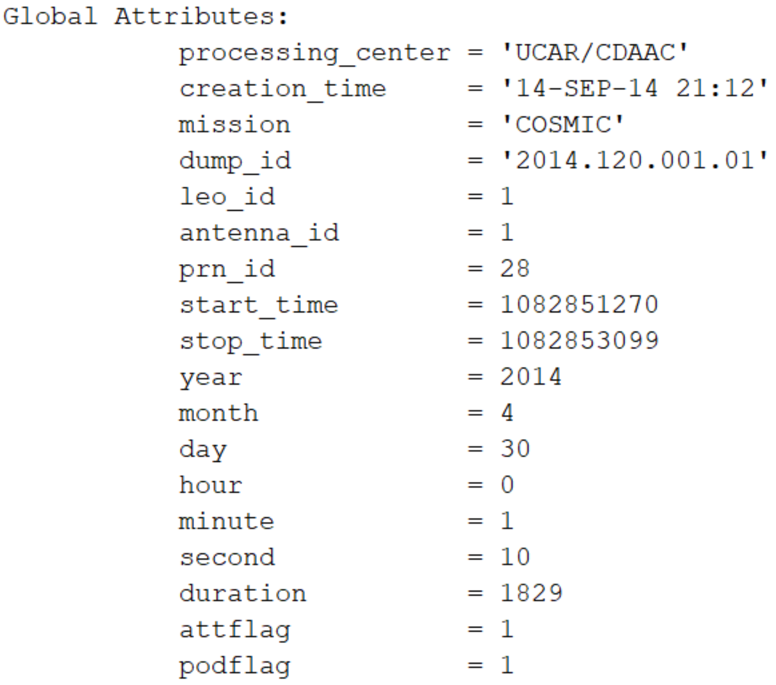
The COSMIC (Constellation Observing System for Meteorology, Ionosphere, and Climate) program is a global leader in the retrieval of GNSS data and largely provides this data for free to be used by the public [7]. Much of this data is widely in active use in research in fields such as weather and space weather. This naturally fits well with this project given that the theme of this project is also to do with space weather.

A wide amount of data is available including data for Ionospheric excess phase, Total Electron Content, Atmospheric profiles, Spacecraft attitude information and much more. For the purposes of this project, we will only be looking at the data for Total Electron Content.

These files are NetCDF (.nc) files which are commonly used for the exchange of scientific data. This file type was also developed by UCAR. These files were also only available to download in bundles consisting of a day's worth of data. Although it is unfavourable that we are not able to have greater control over the amount of data that we can download at a time, this is more manageable than NASA Earth Data.

3.1.5 Total Electron Content File Content and Structure

The content of these files is in podTec format. The content consists of a Global Attributes/Data section,

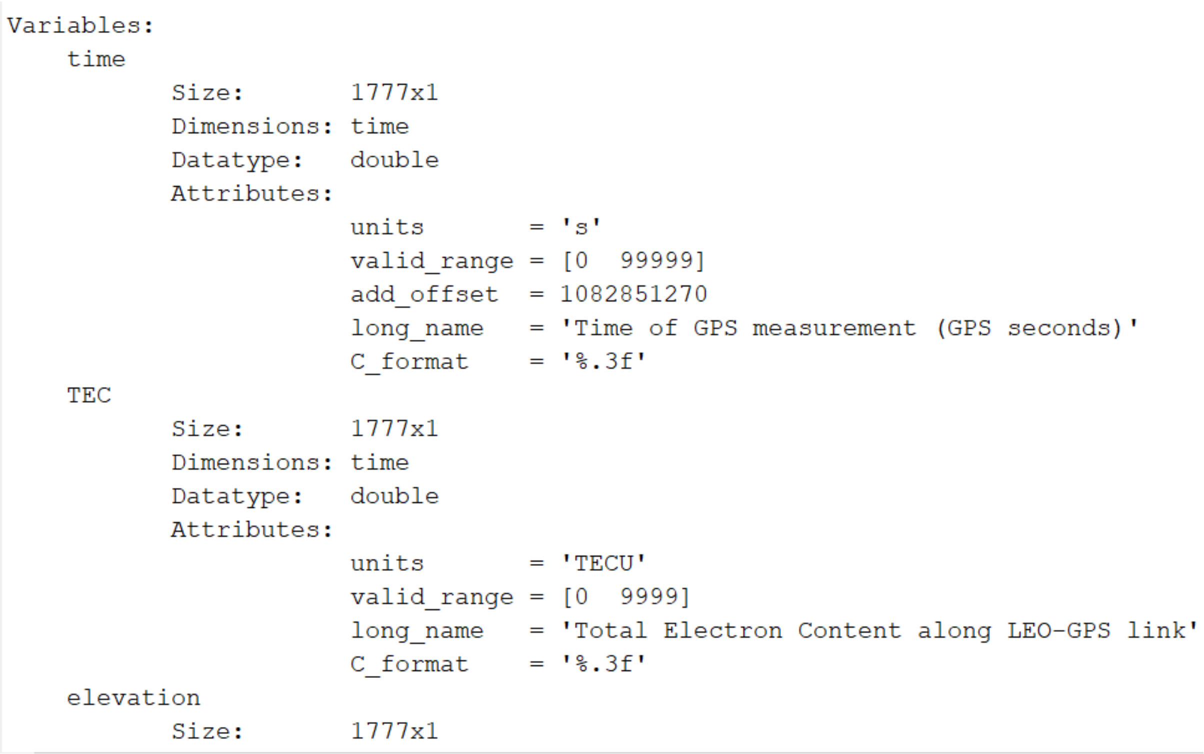


```
Global Attributes:
           processing_center = 'UCAR/CDAAC'
           creation_time     = '14-SEP-14 21:12'
           mission           = 'COSMIC'
           dump_id           = '2014.120.001.01'
           leo_id            = 1
           antenna_id        = 1
           prn_id            = 28
           start_time        = 1082851270
           stop_time         = 1082853099
           year              = 2014
           month             = 4
           day               = 30
           hour              = 0
           minute            = 1
           second            = 10
           duration          = 1829
           attflag           = 1
           podflag           = 1
```

Figure 12: TEC File Contents

Example section of global attributes

and a Variables section.



```
Variables:
    time
           Size:       1777x1
           Dimensions: time
           Datatype:   double
           Attributes:
                       units        = 's'
                       valid_range = [0   99999]
                       add_offset   = 1082851270
                       long_name    = 'Time of GPS measurement (GPS seconds)'
                       C_format     = '%.3f'
    TEC
           Size:       1777x1
           Dimensions: time
           Datatype:   double
           Attributes:
                       units        = 'TECU'
                       valid_range = [0   9999]
                       long_name    = 'Total Electron Content along LEO-GPS link'
                       C_format     = '%.3f'
    elevation
           Size:       1777x1
```

Example section of Variables

Using an appropriate programming language, these values can be read and extracted, and then manipulated and used in any way, as we will be using in our Machine Learning program.

3.2 Selecting the Toolset (Programming Language and Development Environment)

In order to understand and visualise the data, and then implement Machine Learning upon the data, it is necessary to have the correct software toolset available. We must choose a programming language and development environment.

There are many different programming languages and development environments out there to choose from, each with their own advantages and disadvantages. There isn’t necessarily a correct answer or a best language to choose. There are many factors which come into play, a lot of which simply come down to personal preference.

3.2.1 Factors to be Considered

Some of these factors include the experience level of the user, it makes more sense to go with a language with which the user has more experience with as it will be easier to code and debug, saving time and allowing more time to be spent on more important tasks. The ease of use of the language, it’s popularity is also very important. A more popular and widely used language means a better and bigger online community available to help when in need, more answers to common issues, therefore making it a lot easier to debug when stuck. Also, certain languages are better designed for the task at hand. Some languages are better made for data analysis, can crunch numbers more accurately and efficiently.

3.2.2 The Choices

Considering many factors including the ones mentioned above, the options that stood out to me were Python and MATLAB. Both are widely used by the scientific and engineering community for tasks such as data analysis and Machine Learning. Python is famous for being a beginner friendly language, making it advantageous and easier to pick up to those unfamiliar with the language. It is also very popular, with a massive online community and tons of already answered questions and problems, and a large quantity of tutorials available which makes the learning curve less intimidating. Therefore, the initial choice made was to go with Python for this project.

3.2.3 Choosing a Development Environment

Next comes picking a development environment. There are many different ways in which Python programs can be run, and like the earlier problem of choosing a programming language, these development environments also each come with their sets of advantages and disadvantages. However, fortunately in the case of development environments, there are far fewer choices, making it easier and faster to narrow down the best choice for this project. Some of these choices include Jupyter Notebook, Spyder, PyCharm, Visual Studio etc [8].

3.2.4 Jupyter Notebook (and it's issues)

Jupyter Notebook was chosen for this project for the following reasons. It is popular with a large community and following. It has extensive useful such as the ability to run singular lines of code or in batches. Compared to other development environments, it also has better data visualisation capabilities such as graphs and geographical maps and the sort.

Unfortunately, however, after using Jupyter Notebook for a while into the project, many issues starting appearing and the hassle starting to outweigh the advantages. One of the biggest issues was to do with the installation of libraries and external modules. To explain, while Python is beginner friendly and simple, a lot of more complex functions are not natively available and therefore must either be coded from scratch (very difficult and time inefficient) or attained via libraries and external modules. This process of installing modules can at times prove to become quite troublesome. Errors due to internal conflicts between existing and new modules often come up. Errors of this nature can be extremely time consuming and difficult to debug due to the large number of possible causes of the problem. It can often turn into a session of trial and error of numerous recommended fixes available online. At one point in this project, Jupyter Notebook had to be completely uninstalled and then reinstalled for the project to resume, which in the end only served as a temporary fix as similar errors started appearing in a short matter of time!

3.2.5 The Switch to MATLAB

In the end, it was decided to make a switch to MATLAB. Despite it being the language with which I had less experience, it proved to be simple and intuitive enough to pick up enough nicely without sacrificing too much time in the process of switching over. There are less libraries and modules available in MATLAB, making it easier to search through and find the modules that may be desired. However, MATLAB has a lot more functionality built in natively such as functionality to open and read NetCDF files, a feature that wasn't available in Python. These conveniently eliminates the need for many external modules and allows for a smoother development process.

3.3 Data Preparation

3.3.1 Preparing the Data

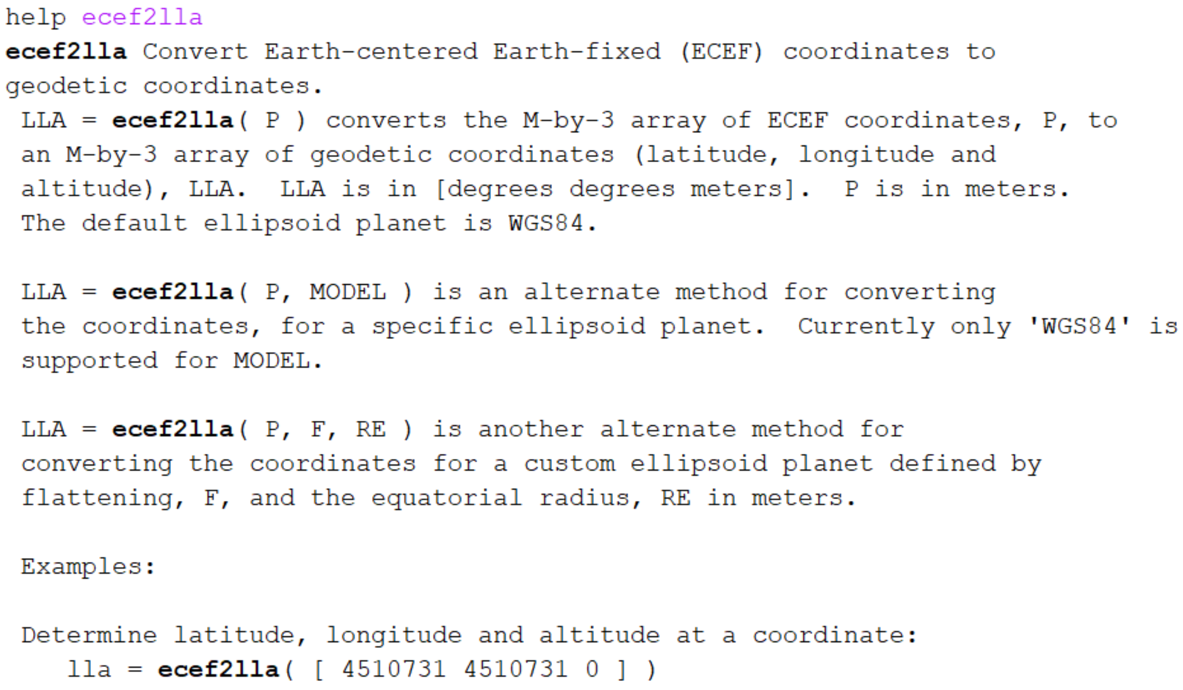
Now that the software toolset has been selected, it is time to put it to use. The first task that comes to attention is that of preparing the data. Data as it is initially downloaded, is not always in the most optimal format to work with, hence the need to prepare / modify it.

3.3.2 ECEF to Geodetic Co-ordinates Example

An example of data that needed preparing in this project is the co-ordinates. The co-ordinates for the LEO satellite in the data is given in ECEF (Earth Centered Earth Fixed) format. This becomes an issue because many of the MATLAB functions which we would like to use when visualising the data expects the input co-ordinates data to be in terms of latitudes and longitudes. Therefore, it is necessary to convert the data from ECEF to geodetic.

3.3.3 Conversion Example

There are many ways to go about preparing data. It can be coded up manually, or if available, MATLAB's built-in functions may be used. In the example of the co-ordinates, MATLAB has a built-in function to help with the conversion. There is a function called ecef2lla which takes in a set of ECEF co-ordinates and returns the equivalent geodetic co-ordinates. Here is a picture of MATLAB's official definition of the function.



```
help ecef2lla
ecef2lla Convert Earth-centered Earth-fixed (ECEF) coordinates to
geodetic coordinates.
 LLA = ecef2lla( P ) converts the M-by-3 array of ECEF coordinates, P, to
 an M-by-3 array of geodetic coordinates (latitude, longitude and
 altitude), LLA.  LLA is in [degrees degrees meters].  P is in meters.
 The default ellipsoid planet is WGS84.

 LLA = ecef2lla( P, MODEL ) is an alternate method for converting
 the coordinates, for a specific ellipsoid planet.  Currently only 'WGS84' is
 supported for MODEL.

 LLA = ecef2lla( P, F, RE ) is another alternate method for
 converting the coordinates for a custom ellipsoid planet defined by
 flattening, F, and the equatorial radius, RE in meters.

 Examples:

 Determine latitude, longitude and altitude at a coordinate:
    lla = ecef2lla( [ 4510731 4510731 0 ] )
```

3.3.4 Directory and File Organisation

Another issue that came up when dealing with data preparation was the way the files were organised. Single files did not contain data for a whole day, one day was compromised over many files and these files contained data from different satellites. Ideally, we want to be looking at one day's worth of data at a time.

The Fix

The data from the files consisting of a single day had to be combined. The approach taken in this project was to first organise the files into directories where each directory had the files for one day. This was further separated by which satellite recorded the data. As an example, if the TEC data were to be combined for a data, an array was first instantiated for the complete TEC data of the day. This may be called TotalTEC for example. Then, a for loop was ran over all the files in the directory, and each of the TEC data within the individual files were added to the TotalTEC variable. In the end the TotalTEC variable would have all the TEC data for a day. This process can be modified to collect data for any time span.

3.4 Data Visualisation

3.4.1 Why Visualise?

The next step to take was to visualise the data. This allows the user to develop a more intuitive feel of the data and to understand it slightly better. MATLAB has many built in functionality which makes it convenient and simple to visualise our data. Some of these options include simple 2d plots, 3d plots, geographic maps with different variations such as bubble plots, colour plots and much more.

3.4.2 What to Visualise: Total Electron Content (TEC)

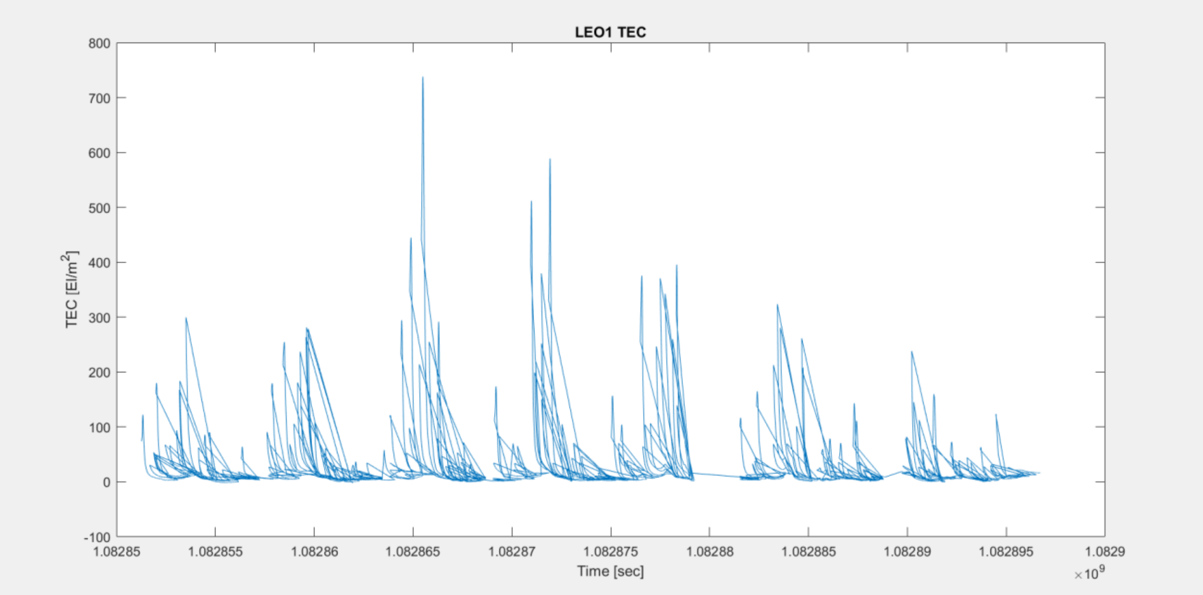
The data that we will primarily be focusing on in this project is the TEC. The file structure has been shown and discussed previously. The total electron content is the number of electrons that are present in a path between a radio transmitter and receiver. It is measured in el/m^2 (electrons per square metre). A TEC unit is represented as TECU which is equal to 10^16 electrons / m^2. The value of the TEC is dependent upon factors such as time, latitude, longitude, and several other geomagnetic conditions [9].

Importance

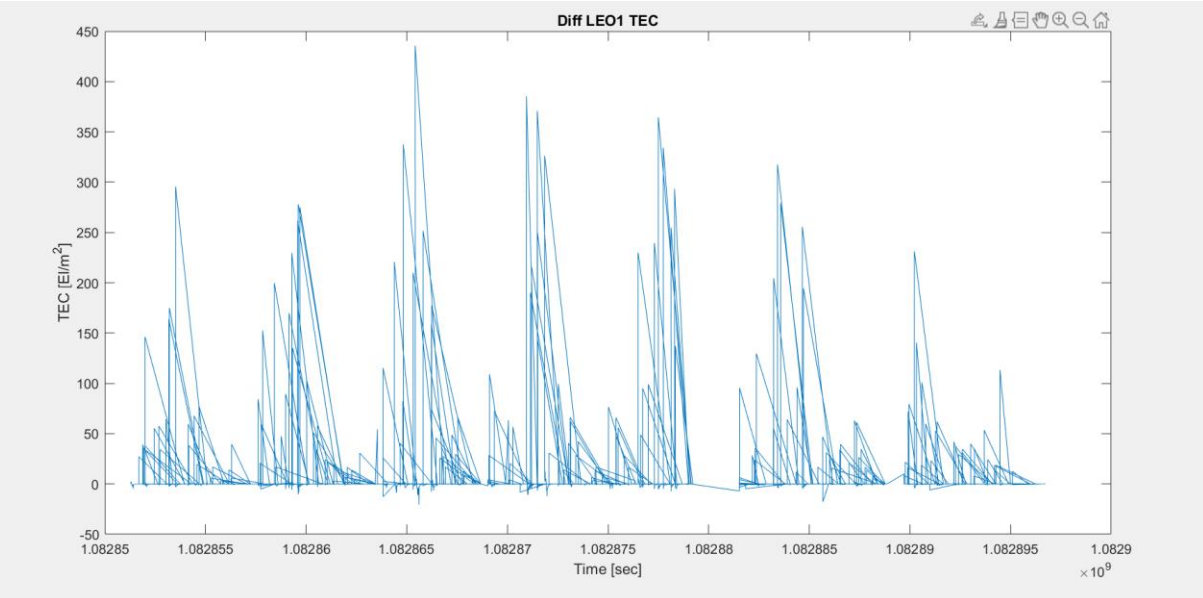
Radio waves are affected by electrons. The more electrons there are, the more the radio waves are disturbed. As the waves propagate through the ionosphere, the electrons can cause the wave's velocity and path to change. This leads to big impacts on the accuracy of satellite navigation systems such as GPS and GNSS. The error caused due to this can be many tens of meters. Therefore, it is worthwhile to study and monitor TEC.

3.4.3 MATLAB Plots

As mentioned earlier, TEC is dependent upon time, latitude and longitude among other factors. We have access to these variables in our data file so it make sense to work with them. We can start by making some plots of TEC vs some of these variables. (Note that in this project, due to hardware restrictions, a limited set of data was used. For more accurate plots, it is recommended to use as much data as possible).



This 2d graph plots the TEC against time that was recorded by LEO 1. (There are multiple LEOs each taking their own recording)



This 2d plot shows diff(TEC) against time. Diff(TEC) is an array of the differences between adjacent elements of the original TEC array.

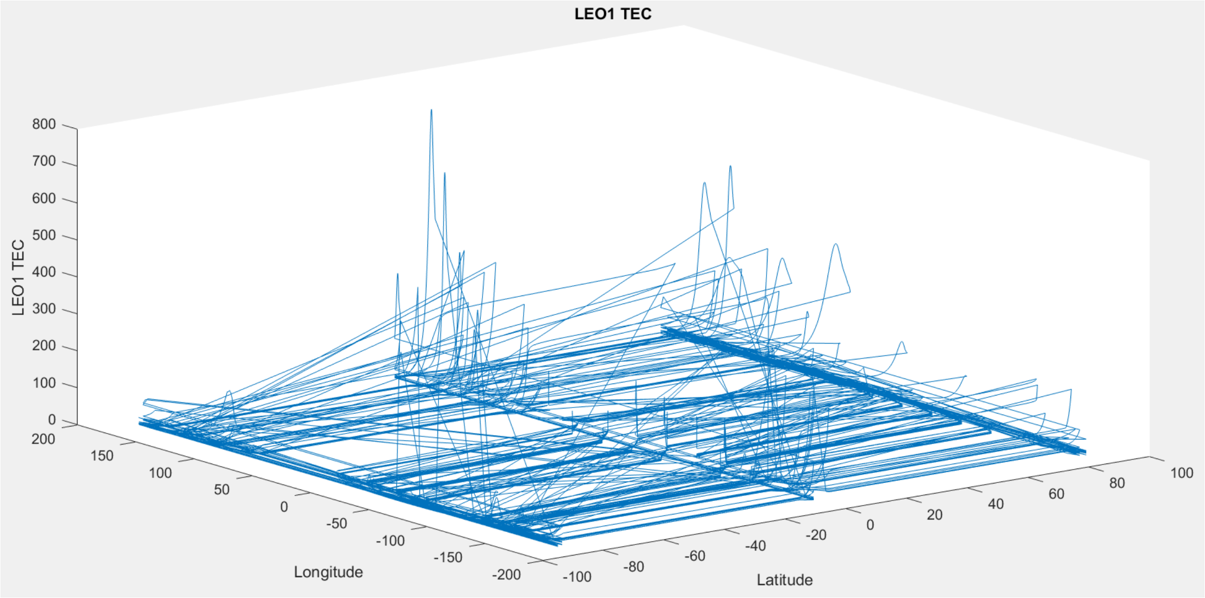


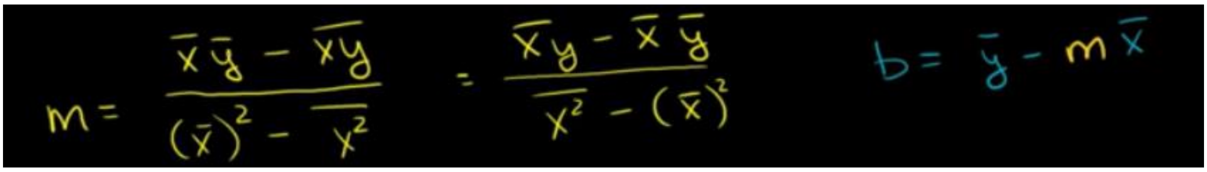
Figure 13: 3d Plot of TEC vs Latitude and Longitude

This is a 3d plot of TEC against longitude and latitude

3.4.4 Machine Learning using Regression

Now we will actually implement some machine learning. For this, we will be using Linear Regression which is a type of supervised learning. Linear regression is used to make estimations for relationships between variables by creating a line of best fit. This line can be used for forecasting by inputting in future values of inputs and getting an estimate for the output.

The line of best fit is of the format y = m*x + b where m is the slope and b is the y-intercept. To calculate values for m and b we use the following simplified formula:



$$m = \frac{\bar{x}\bar{y} - \overline{xy}}{(\bar{x})^2 - \overline{x^2}} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - (\bar{x})^2} \qquad b = \bar{y} - m\bar{x}$$

Figure 14: Linear Regression Equation (Khan Academy)

X bar is the mean of x. Y bar is the mean of y and so on.

Regression for TEC vs Time

Below is a program in MATLAB that has been made to perform linear regression using TEC as the y values and time as the x values:

```
% Directory holding all the data for the LEO 1
LEO1 = dir("LEO1");

% Variables to hold the value of the TEC and Time
```

```
LEO1_TECS = [];
times = [];

% Getting the TEC and times from all the files
% Note the loop is going till 20 due to hardware
speed restrictions
% For the full set of data, replace 20 with
length(LEO1)
for i = 1: 20
    file_name = strcat('LEO1/', LEO1(i).name);
    if (isfile(file_name))
        TEC = ncread(file_name, 'TEC');
        TEC = transpose(TEC);
        LEO1_TECS = [LEO1_TECS, TEC];

        time = ncread(file_name, 'time');
        time = transpose(time);
        times = [times, time];
    end
end

% Set times as x and TEC as y variables
x = times;
y = LEO1_TECS;

noOfX = length(x);
sumX = 0;
sumY = 0;
sumXY = 0;
sumXSquared = 0;


% Get values needed to calculate means
for i = 1: length(x)
    sumX = sumX + x(i);
end

for i = 1: length(x)
    sumY = sumY + y(i);
end

for i = 1: length(x)
```

```
    sumXY = sumXY + x(i)*y(i);
end

for i = 1: length(x)
    sumXSquared = sumXSquared + (x(i)^2);
end

% Get the means of x, y, xy, x^2 etc
meanX = sumX / noOfX;
meanY = sumY / noOfX; % since length of y is =
length of x
meanXY = sumXY / noOfX;
meanXSquared = sumXSquared / noOfX;
squareOfXMean = meanX^2;

% Regression Line
% calculate slope m
m = (meanX * meanY - meanXY) / (squareOfXMean -
meanXSquared);
% calculate slope b
b = meanY - m * meanX;

% Line of best fit using regression
yRegression = m * x + b;

% Plot Graph
plot(x, yRegression);
```

From this line, future values of TEC can be predicted by inserting new unknown values of time.

4 Outcomes / Results

We have done background research, found out what has been done in the past. We search for data and managed to locate and download it. We were then able to clean and prepare the data in a way that made it usable for our project. We explored through available options for our software toolset, made a selection and learned functionalities used for machine learning such as data manipulation, data cleaning, data visualisation, making regression lines and much more. We implemented supervised machine learning through linear regression to our data. All in all, we have managed to accomplish a lot in this project that may serve as a foundation and reference for future projects of this nature.

Thoughts

Attempting to integrate machine learning with space weather has indeed been challenging and there were many times when frustration seemed overwhelming. Unexpected bugs and errors in the code and software which took much more time than anticipated, to issues with hardware limitations which restricted the amount of data we would like to have used, there are many problems that can and have popped up.

However, at the same time, a deeper appreciation for the opportunities and potential of projects of this nature has been developed. For a student with limited resources to be able to freely get their hands on data and software tools, and with the tutorials and help available online through the form of videos, websites and the like, it is clear that many greater and useful things can be accomplished in this field by professionals with more knowledge, experience and resources. It would be worthwhile for more people to invest time and money into research projects like these.

5 Milestones

There were several major time consuming challenges which served as significant milestones for this project. This includes:

Background Research

Learned background knowledge (machine learning and space weather) which are necessary for the project. Located and read through literature and past work of researchers and developed an appreciation of the nature and possibilities of the project tasks.

Data Search and Acquisition

Searched for and located data. Sorted through different formats of data available while considering the practically of sizes of data available. Downloaded and confirmed usability of data.

Software Tools Selection and Set up

Researched available options for programming languages and development environment. Installed and made a selection for the most appropriate one. Set up development environment. Spent time learning and familiarising with the necessary functions of the language and software.

Data Preparation

Prepared and cleaned data in order for it to be usable for the project. Changed data formats to other formats when necessary (edec co ordinates to geodetic).

Data Visualisation

Created multiple graphs visualising relationships between different variables including 2d graphs and 3d plots using MATLAB.

Machine Learning Implementation

Implemented machine learning on the data using linear regression from supervised learning. Coded up and fully commented.

6 Uncertainty / Error

As with any research project, there is almost always some degree of uncertainty and error involved and it is important to address and state this. As the famous professor Walter Lewin of MIT once said, any measurement made without the knowledge of it's uncertainty is meaningless. Therefore, we will take a moment to discuss the possible sources of error in this project.

6.1 Amount of Data used

Due to hardware restrictions during the course of this project, a minimal set of data was used. As machine learning relies heavily upon learning from data, this means that the final results produced by the program were not as accurate as could be. The restricted data set lead to a lower quality training data set. The simple solution to this would be to use more data (which of course requires better and faster hardware).

6.2 Uncertainty in the Data

The data that was used in this project was not perfect by any means. The data provided by measurements from UCAR COSMIC also contains it's own share of error and uncertainty. This error also ripples in and affects the final total error.

6.3 Algorithms and Methods Used

As mentioned previously, there is a huge degree of freedom when it comes to machine learning projects. There are many algorithms to choose from. While theoretically, they should all produce the same results, in actual implementation some do outperform others producing better results. Therefore, it is fair to say that some of the error in the results of this project, some of the error in the predictions made can be attributed to the algorithms and other design choices that we used.

7 Further Recommended Work

There are many different ways in which this project can be further progressed and improved it. As mentioned previously, there is a huge degree of freedom in the nature of such projects, and this project has shown the basics and foundations of starting such a project. Numerous parameters exist to be fiddled and played around with.

7.1 Repeating with Better Hardware

If better faster hardware is available, it may be worthwhile to repeat some of the programs in this project with a larger set of data. Using more data is known to improve the accuracy of machine learning program results. The amount of data that we were able to use was restricted by the speed of our hardware.

7.2 Using Different Inputs

There are many different parameters that affect the TEC value. This project only tried a few of them such as time, longitude and latitude as input parameters to our machine learning program. Why not experiment with other factors and see how the accuracy of the results change? This can be computationally heavy and time consuming however.

7.3 Using a Different Machine Learning Algorithm

There are numerous different machine learning algorithms out there (while this project only implemented one which was linear regression). The same task of forecasting TEC data can be done again but with the implementation of different algorithms such as logistic regression, decision trees, naive bayes, support vector machines and much more. This may change and even improve our results.

7.4 Algorithm Comparison

If the forecasting task has been implemented with different algorithms, then the next logical step may be to compare their results in terms of factors such as accuracy and speed. It may be useful to produce a clear side by side comparison of the pros and cons of different algorithms.

7.5 Forecasting Different Space Weather Variables

While we chose to focus on the TEC variable to measure space weather impacts, there remains many other interesting variables that are vulnerable to the impacts of space weather. They may be dependant upon many factors and a lot of data has been recorded for them over the years (making it an ideal machine learning research task). It may be worthwhile to try and make forecasts for them as well.

8 References

[1] E. Camporeale, “The Challenge of Machine Learning in Space Weather: Nowcasting and Forecasting,” 2019.

[2] E. Camporeale, Machine Learning Techniques for Space Weather, 2018.

[3] M. OpenCourseWare, “Introduction to Machine Learning,” [Online]. Available: https://www.youtube.com/watch?v=h0e2HAPTGF4.

[4] H. J.Berliner, “Backgammon computer program beats world champion,” 1980.

[5] R. Bala, “Real-time prediction of magnetospheric activity using the Boyle Index,” 2009.

[6] NASA, “NASA Earth Data,” [Online]. Available: https://earthdata.nasa.gov/.

[7] “UCAR COSMIC,” [Online]. Available: https://www.cosmic.ucar.edu/.

[8] W. Fatima, “PyCharm Vs Spyder Vs Jupyter: Best Choice For Python Programming,” [Online]. Available: https://ssiddique.info/pycharm-vs-spyder-vs-jupyter.html.

[9] SWPC, “TOTAL ELECTRON CONTENT,” [Online]. Available: https://www.swpc.noaa.gov/phenomena/total-electron-content.

9 Table of Figures

Figure 1: Effects of Impacts of Space Weather 7
Figure 2: Geomagnetically Induced Currents 8
Figure 3: Signals travelling through Ionosphere 9
Figure 4: Supervised Learning 10
Figure 5: Unsupervised Learning 11
Figure 6: Reinforcement Learning 12
Figure 7: Google Search Results for TEC 15
Figure 8: Earth Data Home Page 16
Figure 9: Earth Data Search Results 16
Figure 10: Granules of Data 17
Figure 11: UCAR COSMIC Home Page 17
Figure 12: TEC File Contents 18
Figure 13: 3d Plot of TEC vs Latitude and Longitude 24
Figure 14: Linear Regression Equation (Khan Academy) 24